

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

VIŠJE RAČUNSKE METODE

4. naloga: Kvantni mnogodelčni problemi: kubitne verige

Žiga Šinigoj, 28222025

Ljubljana, marec 2023

1 Izračun

V nalogi je potrebno implementirati propagacijo (e^{zH}) po času $z = -it$. Če je število spinov sodo lahko lahko Heisenbergov Hamiltonijan razbijemo na 2 člena, ki med sabo komutirata (sodi in lihi del). Ker operatorji na sodih mestih komutirajo sami med sabo (enako za lihi del) jih lahko zapišemo kot produkt eksponentov. Ob uporabi split-step metode se propagacija poenostavi na delovanje vseh sodih členov na stanje in delovanje vseh lihih členov na stanje izmenično. Delamo v lastni bazi operatorja σ_j^z , spin 0 pomeni spin "gor", spin 1 pa spin "dol". Dovolj je pogledati, kako deluje en operator iz produkta sodih operatorjev (ali lihih) na stanje. Dvodelčni operator je tako

$$U_{j,j+1}^{(2)} = e^{z\vec{\sigma}_1 \cdot \vec{\sigma}_2} = e^{-z} \begin{pmatrix} e^{2z} & 0 & 0 & 0 \\ 0 & \cosh(2z) & \sinh(2z) & 0 \\ 0 & \sinh(2z) & \cosh(2z) & 0 \\ 0 & 0 & 0 & e^{2z} \end{pmatrix} \quad (1)$$

, kjer indeks j označuje spin na mestu j . Sodi člen Hamiltonijana bo sedaj produkt dvodelčnih operatorjev, ki bodo delovali na mestih $U_{2j,2j+1}^{(2)}$. Podobno za lihi člen Hamiltonijana. Vsak dvodelčni operator lahko deluje na stanja ali pa na koeficiente. Bolj prikladno je da deluje na vektor koeficientov, kjer vsak kompleksni koeficient v vektorju predstavlja eno od stanj baze. Torej izbrani dvodelčni operator spremeni koeficiente v vektorju velikosti $N = 2^n$, kjer je n število spinov. Pričakovano vrednost operatorja, korelacijsko funkcijo, particijsko in avtokorelacijsko funkcijo spinskega toka dobimo po naslednjih formulah

$$Z(\beta) = \frac{1}{N_\psi} \sum_{|\psi\rangle} \langle \psi(-i\beta/2) | \psi(-i\beta/2) \rangle, \quad (2)$$

$$\langle X \rangle_\beta = \frac{1}{Z(\beta)} \frac{1}{N_\psi} \sum_{|\psi\rangle} \langle \psi(-i\beta/2) | X | \psi(-i\beta/2) \rangle, \quad (3)$$

$$\langle X(t) | X(0) \rangle = \frac{1}{N_\psi} \sum_{|\psi\rangle} \langle \psi | \exp(itH) X \exp(-itH) | \psi \rangle \quad (4)$$

$$J = \sum_{i=1}^N J_i, \quad (5)$$

$$J_i = \sigma_i^x \sigma_{i+1}^x - \sigma_i^y \sigma_{i+1}^y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2i & 0 \\ 0 & -2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6)$$

$$\langle J(t) J(0) \rangle = \sum_{r=0}^{n-1} \frac{1}{N_\psi} \sum_{|\psi\rangle} \langle \psi | J_{r+1}(t) J_1(0) | \psi \rangle. \quad (7)$$

Če ni označeno drugače je $N_\psi = 6$.

2 Implementacija: delovanje operatorja na stanje sistema

Najprej je potrebno implementirati dvodelčni operator na stanje. Kot sem že omenil sem poljubno stanje zapisal kot vektor kompleksnih koeficientov baznih stanj $\vec{c} = (c_0, c_1, \dots, c_N)$. Indeksi koeficientov predstavljajo mesta, kjer se nahajajo v vektorju. Če sedaj prepišem indekse v binarni sistem lahko implementiram dvodelčni operator. Koeficient c_0 pripada stanju sistema $|00\dots 0\rangle$, koeficient c_1 pripada baznemu stanju $|00\dots 01\rangle$ itd. Operator $U_{i,i+1}^{(2)}$ deluje na vse koeficiente, glede na stanje bitov na mestih $|i, i+1\rangle \in |00\rangle, |01\rangle, |10\rangle, |11\rangle$, npr. $U_{0,1}^{(2)}$ bo transformiral koeficient $c_0 \rightarrow e^z c_0$, koeficient c_1 pa kot $c_1 = c_{|0\dots 01\rangle} \rightarrow e^{-z} \cosh(2z) c_1 + e^{-z} \sinh(2z) c_{|0\dots 10\rangle}$, kjer sem koeficient $c_{|0\dots 10\rangle}$ poiskal tako, da sem zamenjal bita na mestih $i, i+1$ in pretvoril spet v decimalno obliko. Na ta način sem dobil indeks iskanega koeficienta v vektorju. Ko poznamo delovanje dvodelčnega operatorja dobim delovanje sodega (lihega) dela Hamiltonijana kot zaporedno delovanje dvodelčnih operatorjev na vseh sodih (lihih) parih spinov. Predpostavimo tudi periodične robne pogoje.

Na podoben način sem implementiral Hamiltonijan na stanje sistema. Dvodelčni blok v Hamiltonijanu, ki deluje na bita $i, i + 1$ je

$$\vec{\sigma}_i \cdot \vec{\sigma}_{i+1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)$$

Preprosto je delovanje operatorja σ_i^z , saj smo v lastni bazi. Koeficienti se transformirajo glede na vrednost i -tega bita. Če je na mestu i spin "gor" potem se koeficient, ki pripada danemu stanju ne spremeni. Če je na mestu i spin "dol" potem pomnožimo koeficient z (-1) .

3 Rezultati

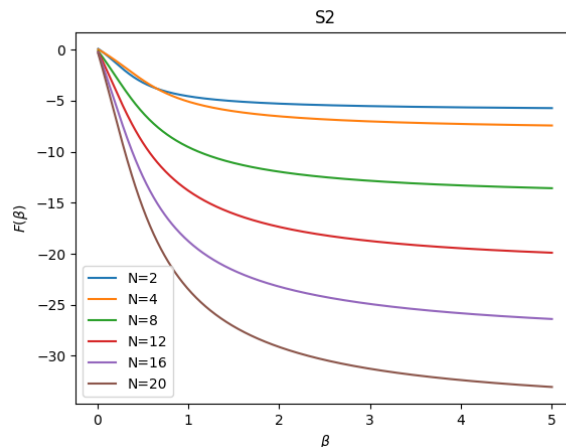
Izračune sem naredil v C++, grafe pa v Pythonu. Za test če program pravilno deluje lahko na roke diagonaliziramo matriko za dva kubita in najmanjša lastna vrednost da energijo osnovnega stanja. Torej v limiti majhnih temperatur moramo dobiti $\lim_{\beta \rightarrow \infty} \langle H^{(2)} \rangle = -6$. Izračun energije za $n = 2$ se sklada z teorijo (slika 1). Kljub temu pa sem imel kar nekaj težav z unitarnostjo sistema, kar mogoče implicira, da je kakšna napaka v programu. Propagacija sistema je unitarna operacija, vendar s časom stanje sistema ne ohranja norme do reda, ki ga podajajo split-step formule. Ker vem, da je propagator unitaren, bi mogoče lahko ob vsakem koraku v času normalizirati vektor koeficientov. Težava se pojavi predvsem pri avtokorelacijskih funkcijah, saj lahko te grejo proti nič kot posledica odstopanja propagatorja od unitarnosti, kar lahko povzroči napačno interpretacijo fizike. Kljub temu je oblika avtokorelacijskih funkcij enaka če normiram stanje ali pa ne, razlika je predvsem v amplitudi.

Vseeno mi ni dalo miru, da split-step sheme ne ohranjajo norme vektorja koeficientov. Izkazalo se je, da je ob delovanju dvodelčnega operatorja na stanje v primeru $|01\rangle$ oz. $|10\rangle$ spremenilo koeficient vektorja, katerega vrednost sem uporabil za izračun na stanju $|10\rangle$ oz. $|01\rangle$ (pomembno je, da se čimbolj izogibam kopijam vektorja stanja za hitrejšo delovanje kode). Vseeno kljub napačni implementaciji sem dobil pravi izračun energij sistema (najverjetneje zaradi ansambelskega povprečja stanj) in tudi korelacijske funkcije niso veliko odstopale, razen seveda zmanjševanja amplitude zaradi neunitarnosti sistema. Ponovil sem izračune za korelacije, odstopanja energij in avtokorelacijsko funkcijo spinskega toka. Če ne bi slutil, da morajo split-step formule bolje ohranjati normo (čeprav jo tudi te vedno slabše z večanjem sistema) ne bi mogel vedeti, da je implementacija napačna.

Po odpravi napake sem uporabljal split-step metodo S2.

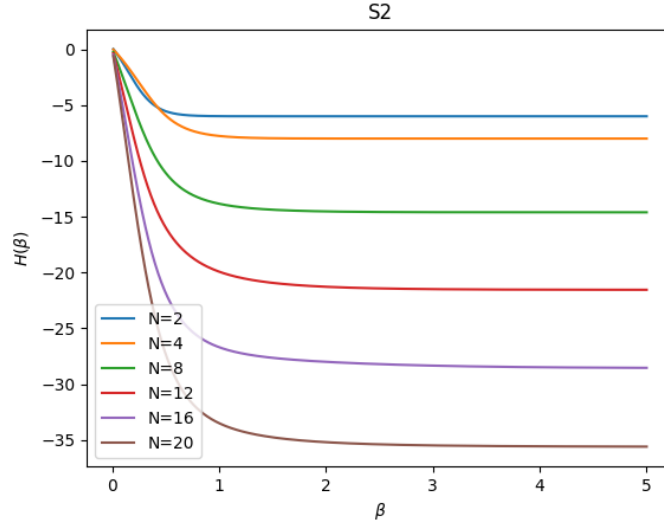
3.1 Prosta in povprečna energija

Prosto in povprečno energijo sistema v odvisnosti od β pri različnih velikostih verige prikazuje slika 1 in 2. Skleпам lahko da v termodinamski limiti velja $F\beta \xrightarrow{\beta \rightarrow \infty} -\infty$, podobno velja tudi za energijo. Z večanjem števila členov v verigi je energija bolj negativna, ker imamo anti-feromagnet.

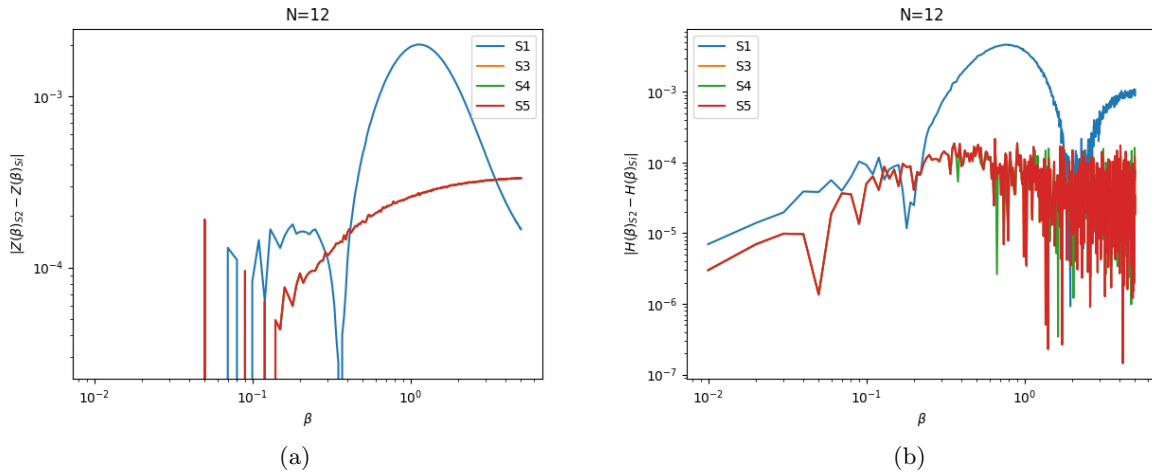


Slika 1: Prosta energija sistema v odvisnosti od inverzne temperature pri različnih velikostih verige spinov.

Da je dovolj split-step metoda S2 nam potrdijo tudi odstopanja različnih metod prikazana na sliki 3.



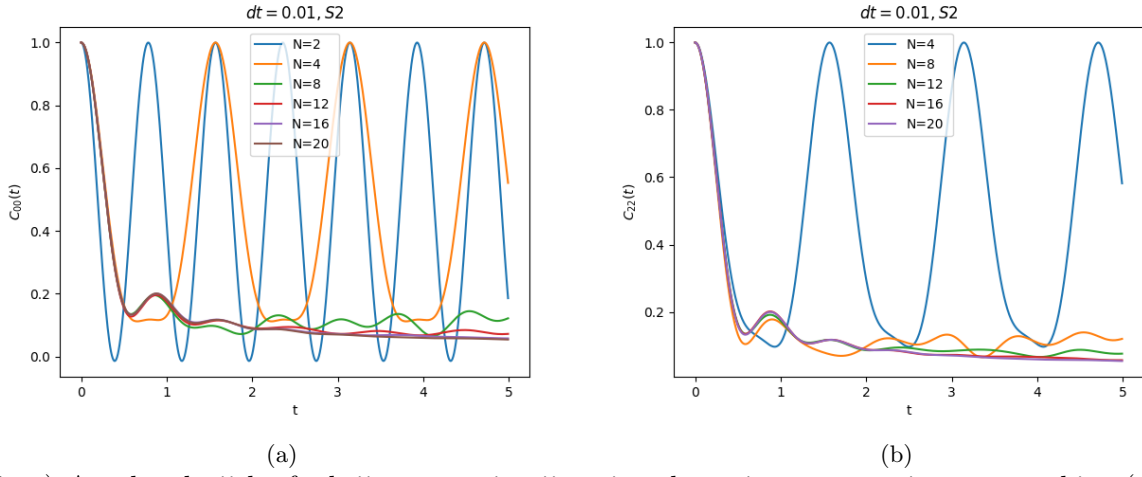
Slika 2: Energija osnovnega stanja sistema v odvisnosti od inverzne temperature pri različnih velikostih verige spinov.



Slika 3: a) Odstopanja metod pri izračunu proste energije. b) Odstopanja metod pri izračunu energije.

3.2 Avtokorelacijska funkcija lokalne magnetizacije

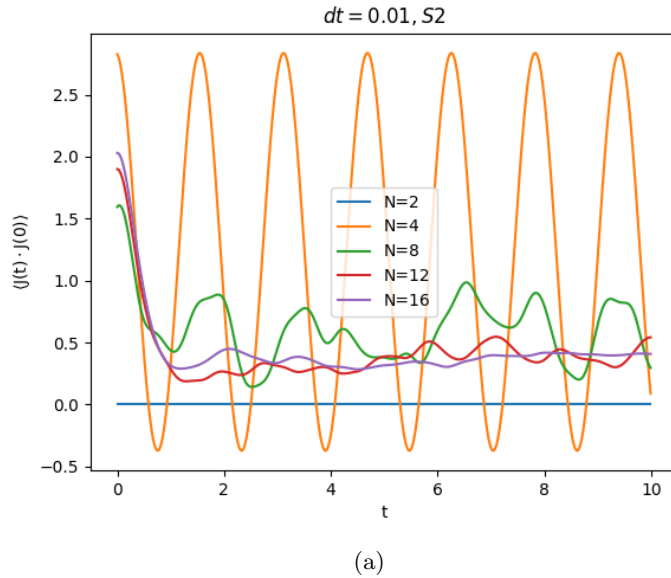
Avtokorelacijsko funkcijo lokalne magnetizacije prikazuje slika 4. Pri $N = 2$ je funkcija periodična, z večanjem števila spinov pa avtokorelacija s časom pada. Tudi periodičnost se manjša z večanjem števila spinov. Za $N \rightarrow \infty$ bi mogoče lahko trdil da ostanejo samo prvi trije vrhovi v avtokorelacijski funkciji. Bi pa potreboval boljši računalnik za preverbo. Avtokorelacijska funkcija na 3. spinu ima podobno odvisnost kot na 1. spinu, večje razlike so samo pri $N = 4$.



Slika 4: a) Avtokorelacijska funkcija magnetizacije pri neskončni temperaturi na prvem bitu ($C_{00}(t)$). b) Avtokorelacijska funkcija magnetizacije pri neskončni temperaturi na tretjem bitu ($C_{22}(t)$).

3.3 Avtokorelacijska funkcija spinskega toka

Podobno kot za avtokorelacijo lokalne magnetizacije ponovimo izračun za spinski tok. Razlika je v tem, da imamo še vsoto po spinih, kar poveča čas računanja za faktor N (dolžina verige). Zaradi velike časovne zahtevnosti sem izračunal samo do $N = 16$. Že za $N = 16$ je bilo potrebno več kot 10h računanja na enem jedru. Kode nisem paraleliziral, saj je to malce bolj zahtevno v C++ in bi potreboval več časa. Spinski tok je nič za dva spina, potem dobim periodično obnašanje za $N=4$, za večje število spinov pa periodičnost izgine in imamo padec spinskega toka ob začetnih časih in nato približno konstantno vrednost za velike N . Ko sem naredil graf pri napačno napisani kodi je spinski tok padal proti nič, kar je pomenilo, da sem lahko določil difuzijsko konstanto. Padanje ni bilo fizikalno ampak posledica neunitarnosti propagatorja. Ob popravljenem programu pa to ne drži in vsaj za dane čase spinski tok ne pada proti nič.



Slika 5: Avtokorelacijska funkcija spinskega toka pri različnih velikostih verige (N).

3.4 Kubo-jeva spinska difuzijska konstanta

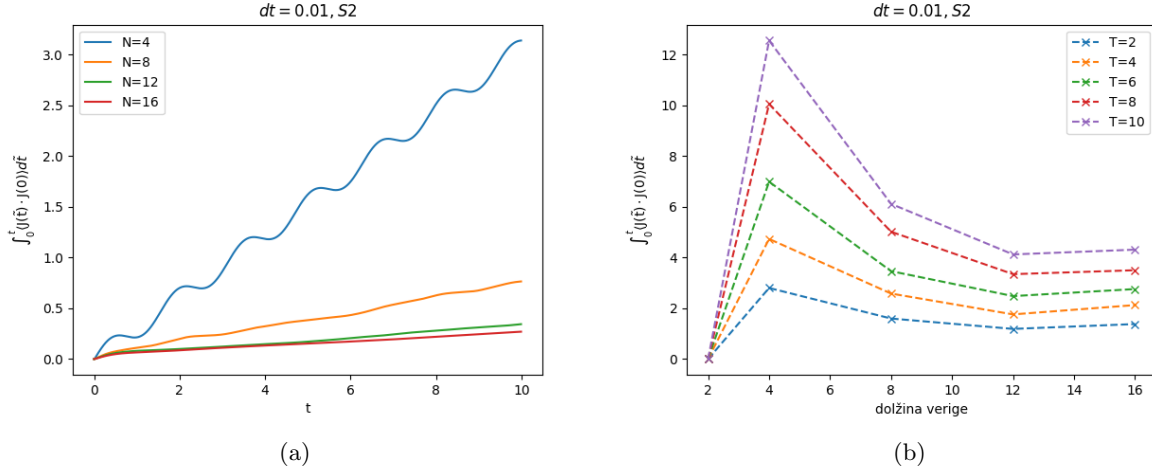
Kubo-jevo difuzijska konstanta je definirana kot

$$D(N) = \lim_{t \rightarrow \infty} \int_0^t \langle J(t') J(0) \rangle dt' . \quad (9)$$

Glede na spinski tok ne zgleda, da bi integral konvergirala. Vseeno pa spinski tok pada z N , torej bi lahko dobil difuzijsko konstanto za neskončno verigo kot

$$D = \lim_{N \rightarrow \infty} \lim_{t \rightarrow \infty} \int_0^t \langle J(t') J(0) \rangle dt' . \quad (10)$$

Integral spinskega toka to časa T prikazuje slika 6a), vidimo lahko da se naklon manjša z večanjem spinov. Če pogledam sedaj še odvisnost od dolžine verige (slika 6b)) vidim, da bi mogoče lahko obstajala konvergenca integrala. Za bolj natančno oceno bi potreboval paralelizacijo in zmogljivejši računalnik.



Slika 6: a) Avtokorelacijska funkcija spinskega toka pri različnih velikostih verige (N). (b) Avtokorelacijska funkcija spinskega toka pri različnih časih (T).

4 Zaključek

Zaradi računalniške aritmetike je relativno preprosto simulirati spinske (kubitne) verige, saj ni potrebno konstruirati celotne matrike velikosti Hilbertovega prostora, ampak samo poiskati neničelne elemente matrike, ki predstavlja dani operator. To močno zmanjša število operacij in posledično čas računanja.